

GUIA DÍA 3

- Crear TABS

```
tabs = dbc.Row(html.Div([
    dcc.Tabs(
        id="tabs-dash",
        value='tab1',
        children=[
            dcc.Tab(label='Información General', value='tab1',
style={'display': 'flex', 'alignItems': 'center', 'justifyContent':
'center'})),
            dcc.Tab(label='Otra Información', value='tab2',
style={'display': 'flex', 'alignItems': 'center', 'justifyContent':
'center'})),
        ])
```

- Añadir tabs al callback

```
@app.callback(
    Output("contenido", "children"),
    Input('selector-ccaa', 'value'),
    Input("tabs-dash", "value")
)
def update_content(value, tab):
    if value is None:
```

para cada una de las tabs:

if tab == "tab1":

 contenido =

 return contenido

elif tab == "tab2":

 contenido =

 return contenido

- Crear botones de descarga de Excel y informe

```
menu = dbc.Row([
    dbc.Col(html.H1("Ventas y Desembarques por CCAA", style={"textAlign":
"center", "color": "#000000", "font-size": 40}), width = 9),
    dbc.Col(se.selector_ccaa(), width= 3),
    dbc.Col([
        dbc.Button("Descargar Excel", id="boton-descarga-excel",
color="primary", outline=True, n_clicks=0),
        dcc.Download(id="descargar-datos-excel"),
    ])
```

```

        dbc.Button("Descargar Informe", id="boton-descarga-informe",
color="primary", outline=True, n_clicks=0),
        dcc.Download(id="descargar-datos-informe")
    ], width=2,
    style={
        "display": "flex",
        "alignItems": "center",
        "justifyContent": "space-around",
        "marginLeft": "40px",
        "marginRight": "0px",
        "padding": "0px",
        "marginTop": "10px"
    })
])

contenido = html.Div(dbc.Row(id= "contenido"))

```

- Adaptar funciones a GET_DATA:
- Get_data_buques

```

def get_data_buques(value, puerto = None):
    data = f.consulta("SELECT * FROM
SGP_CUADROSMANDO.cm.historico_flota")
    data = data[data["IdCcaa"] == value]

    if puerto:
        data = data[data["puerto"] == puerto]

    data = data.groupby("año")["buques"].sum().reset_index()

    return data

```

- Get data Especies

```

def get_data_especies(value, puerto = None):
    data= f.consulta("SELECT * FROM SGP_CUADROSMANDO.cm.ccaa_desembarques
WHERE año = 2024")

    data = data[data["idccaa_base"] == value]

    if puerto:
        data = data[data["PuertoBase"] == puerto]

    data = data.groupby(['Especie'])[["Peso",
"valor"]].sum().reset_index().sort_values(by= "Peso", ascending = False)

    return data

```

- GET Data Provincias

```
def get_data_provincias(value, puerto= None):

    data = f.consulta("SELECT * FROM
SGP_CUADROSMANDO.cm.ccaa_desembarques")
    data = data[data["Año"] == 2024]
    data = data[data["idccaa_base"] == value]

    if puerto:
        data = data[data["PuertoBase"] == puerto]

    data =
data.groupby("ProvinciaDesembarque")["valor"].sum().reset_index()

    return data
```

- Crear Callback Descarga Excel

```
@app.callback(
    Output("descargar-datos-excel", "data"),
    [Input("boton-descarga-excel", "n_clicks"),
     Input("selector-ccaa", "value"),
     Input("tabla-desembarques", "selected_rows"),
     State("tabla-desembarques", "data")],
    prevent_initial_call=True
)
```

- Función de descarga Excel

```
def download_excel(n_clicks, ccaa, selected_rows, table_data):
    if not n_clicks:
        return None

    if not ccaa:
        return None

    if not selected_rows or len(selected_rows) == 0:
        puerto = None

    else:
        row_index = selected_rows[0]
        puerto = table_data[row_index]["PuertoBase"]

    data_buques = gb.get_data_buques(ccaa, puerto)
    data_especies = ge.get_data_especies(ccaa, puerto)
    data_provincias = gp.get_data_provincias(ccaa, puerto)
    data_desembarques = tb.get_data_desembarques(ccaa, puerto)
```

```

dict_data = {
    "Buques": data_buques,
    "Especies": data_especies,
    "Provincias": data_provincias,
    "Desembarques": data_desembarques
}

output_excel = io.BytesIO()
with pd.ExcelWriter(output_excel, engine="xlsxwriter") as writer:
    for key, df in dict_data.items():
        df.to_excel(writer, sheet_name= key, index=False, startrow=1,
startcol=1)

writer.book.close()
output_excel.seek(0)

return (dcc.send_bytes(output_excel.read(), filename=f"Datos.xlsx"))

```

- Probar Botón EXCEL
- Crear Callback informe HTML

```

@app.callback(
    Output("descargar-datos-informe", "data"),
    [Input("boton-descarga-informe", "n_clicks"),
    Input("selector-ccaa", "value"),
    Input("tabla-desembarques", "selected_rows"),
    State("tabla-desembarques", "data")],
    prevent_initial_call=True
)
def download_informe(n_clicks, ccaa, selected_rows, table_data):
    if not n_clicks:
        return None

    if not ccaa:
        return None

    if not selected_rows or len(selected_rows) == 0:
        puerto = None

    else:
        row_index = selected_rows[0]
        puerto = table_data[row_index]["PuertoBase"]

```

```

informe_nacional = inf.generar_informe(ccaa, puerto)

output_informe = io.BytesIO(informe_nacional.encode("utf-8"))
output_informe.seek(0)

return dcc.send_bytes(output_informe.read(), filename=f"Informe de
caracterización.html")

```

-Crear informe.py en “utils”

Imports:

```

import sys
import os
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__),
'..')))

import pandas as pd

from moduls import tabla_desembarques as tb #type: ignore
from moduls import graf_especies as ge #type: ignore
from moduls import graf_provincias as gp #type: ignore
from moduls import graf_buques as gb #type: ignore

import functions as f #type: ignore

```

- Función generar informe:

```

def generar_informe(ccaa, puerto):
    nombre_ca = f.consulta(f"SELECT Descripcion FROM SGP_SIPE.cat.CCAA
WHERE Id = {ccaa}").squeeze()

    if puerto is not None:
        titulo = puerto
    else:
        titulo = nombre_ca

```

- Modificar funciones

Get_data_especies(): añadir argumento informe = False

```

if informe:
    data = data.nlargest(10, "Peso")

```

graf_especies() añadir argumento html = False

```
fig_pie = dcc.Graph(figure= pie, id= "pie-especies-fig", style=
{"width": "100%", "height": "100%"})

if html == True:
    fig_pie = pie.to_html(full_html = True)

return fig_pie
```

grad_buques() añadir html = False

```
if html == True:
    contenido = line.to_html(full_html = True)

return contenido
```

Hacer lo mismo en provincias

- Traer datos en bruto

```
# Datos en bruto
data_buques = gb.get_data_buques(ccaa, puerto)
data_especies = ge.get_data_especies(ccaa, puerto, True)
data_provincias = gp.get_data_provincias(ccaa, puerto)
data_desembarques = tb.get_data_desembarques(ccaa, puerto)
#-----#

# Datos html
buques_html = data_buques.to_html(index= False)
especies_html = data_especies.to_html(index= False)
provincias_html = data_provincias.to_html(index= False)
desembarques_html = data_desembarques.to_html(index= False)
#-----#

# graficos html
graf_buques= gb.variacion_buques(ccaa, puerto, True)
graf_especies = ge.graf_especies(ccaa, puerto, True)
graf_provincias = gp.provincias_desembarque(ccaa, puerto, True)

#-----#
```

- Contenido HTML
- CSS

```
contenido_html = f"""
```

```
<!DOCTYPE html>
<html lang = "es">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Informe de: {titulo}</title>

    <style>
    @page {{
      size: A4;
      margin: 20mm;
    }}

    @media screen {{
      html, body {{
        width: 100%;
        height: 100%;
        display: flex;
        justify-content: center;
        align-items: center;
        margin: 0;
        padding: 0;
        background-color: #f4f4f4;
      }}

      .wrapper {{
        display: flex;
        justify-content: center;
        width: 100%;
      }}

      .container {{
        width: 900px;
        min-height: 1100px;
        background: white;
        padding: 40px;
        margin: 20px;
        border-radius: 8px;
        box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
      }}
    }}

    @media print{{
      body {{
        font-size: 11px;
      }}
    }}
```

```
body {{
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
  color: #333;
  font-size: 15px;
}}

.container {{
  max-width: 100%;
  margin: 0 auto;
  background: #fff;
  padding: 20px;
  border-radius: 8px;
  box-sizing: border-box;
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
  margin: auto;
}}

h1 {{
  text-align: center;
  border-bottom: 3px solid #8db9df;
  padding-bottom: 10px;
  margin-bottom: 10px;
  color: #003c70;
}}

h2, h3, h4 {{
  color: #488cc6;
}}

h2 {{
  border-bottom: 3px solid #cfe3f8;
}}

h3 {{
  margin-left: 20px;
}}

h4 {{
  margin-left: 40px;
}}

.section {{
  margin-bottom: 25px;
  padding-bottom: 15px;
  border-bottom: 2px solid #ddd;
```



```

    }}

    p, ul {{
        line-height: 1.6;
    }}

    ul {{
        padding-left: 40px;
        list-style-type: circle;
    }}

    table {{
        width: 100%;
        border-collapse: collapse;
        margin-top: 10px;
    }}

    th, td {{
        padding: 10px;
        text-align: left;
        border: 1px solid #ddd;
    }}

    th {{
        background-color: #0073e6;
        color: white;
        font-weight: bold;
        text-align: left !important;
    }}

    tr:nth-child(even){
        background-color: #f7f7f7;
    }

    .highlight {{
        font-weight: bold;
        color: #d9534f;
    }}
</style>
</head>

```

- HTML

```

<body>
    <div class = "container">
        <h1>Informe de caracterización de: {titulo}</h1>

```

```

        <div class = "section">
            <h2>1. Análisis de la de flota: {titulo}</h2>
            <h3>1.1 Evolución del Nº de buques:</h3>
            <p>{buques_html}</p>
            <p>{graf_buques}</p>
        </div>

        <div class = "section">
            <h2>2. Peso y valor desembarcados</h2>
            <p>{desembarques_html}</p>
        </div>

        <div class="section">
            <h2>3. Especies principales:</h2>
            <div style="display: flex; gap: 10px;">
                <div style="flex: 1;">
                    {especies_html}
                </div>
                <div style="flex: 1;">
                    {graf_especies}
                </div>
            </div>
        </div>

        <div class = "section">
            <h2>4. Provincias de desembarque:</h2>
            <p>{provincias_html}</p>
            <p>{graf_provincias}</p>
        </div>
    </body>
</html>
"""

return contenido_html

```